Principles of Software Engineering Fall 2021

*[FAU: CEN 4010]*

Milestone 4 Beta Launch and Reviews

**Project Name: Peace of Mind**

Team Name: *Group 03*

Team Lead: Juan Idrovo

**Team Members and Roles:**

**Product Owner and Scrum Master/GitHub Master:**

Ryan Milrad - [rmilrad2016@fau.edu](mailto:rmilrad2016@fau.edu)

**Lead Frontend:**

Juan Pablo Idrovo - [jidrovo2019@fau.edu](mailto:jidrovo2019@fau.edu)

**Lead Backend:**

Mauricio Retana Rodriguez - [mretanarodri2018@fau.edu](mailto:mretanarodri2018@fau.edu)

**Frontend Support:** Armaghan Ali - [alia2019@fau.edu](mailto:alia2019@fau.edu), Ryan Bharath - [rbharath2018@fau.edu](mailto:rbharath2018@fau.edu), Joseph Brown - [josephbrown2020@fau.edu](mailto:josephbrown2020@fau.edu)

**Backend Support:** Ryan Milrad, Ryan Bharath

November 7, 2021

**Revision History**

| Change # | Date changed | Summary |
| --- | --- | --- |
| 00 | 09-25-2021 | Creation of the document |
| 01 | 10-20-2021 | High Level Functional Requirements - Removed Entertainment Ratings  Data Definition - Removed Entertainment features |
| 02 | 10-21-2021 | High Level Functional Requirements - Added more to the Administrator section, made changes to the Browse Website section.  Added Like/dislike functionality.  Added Search Bar that allows user to look for blog posts.  Non Functional Requirements - Minor change to the performance attribute |

# 1 Product Summary

Peace of Mind is a blog site that allows users to view blog posts and connect with other users. During these hard times, it is difficult to meet up with friends and family in person. To make our users feel at home, we designed a website that does just that. People from all over the world will have access to using our website. You can create an account to view the posts and comment on them.

To market the product, we will pick from a few people to have exclusive access to our website and give us valuable feedback to make it better. We can also have an advertisement to expose more potential customers to our product and have them share with their friends and family.

### 1.1 Name of Product

Peace of Mind

### 1.2 Explicit list of ALL Major Committed Functions (Priority 1)

1. Account Creation
   1. The user can create an account to be a member of the website so that they can take advantage of the features that the website has.
2. About Us Page
   1. The user can navigate to this page to view the members who created the website.
3. Edit Profile
   1. The user can edit their profile where they can change their profile picture, name, contact information and biography.
4. Profile Page
   1. The user can now see their own and other peoples’ post and comments on Peace of Mind by going into their profile page. The user is also able to access those posts by clicking on the URL link.
5. Covid Info Page
   1. The user can access the covid information page where they can read up on information about Covid-19. They are also able to view the amount of cases and deaths so far.
6. Website Browsing
   1. The user can navigate to several pages on the website (About Us. Covid Information, Blog categories, Blog Post, Edit Profile pages).
7. Comments
   1. The user can view the blog post and make a comment on it.The comment is viewable by other users.
8. Blog Page
   1. The user can navigate through the blog page where they can see what blogs has been posted and what blogs they can view.
9. Administrative Rights
   1. The administrator is allowed to create new blog categories and new blog posts. They can also grant other users administrative rights.
   2. The administrator can create new blog categories to keep blog posts more organized
   3. The administrator can edit blog posts and edit blog categories
   4. The administrator can delete the user’s account upon request
10. Maintenance of the Website
    1. The owner of the website will be in charge of maintaining the webpage.
11. Category Page
    1. The user can view the category page and look at what they want to view. For example, if the user is interested in sports, they can take a look at the sports category
12. Like/Dislike Post
    1. Registered users can click the like or dislike button, and the number of persons that liked or disliked the same post will be displayed.

### 1.3 Product URL

<https://www.peaceofmind-cen4010.xyz/>

# 2 Usability Test Plan

### 2.1 Test Objective

1. Once clicking onto the section of the menu bar titled Blog Post the user is brought to a new page where they can view what the other users have posted and comment on it.

2. There is a comment section on the blogs made by other users, this allows for anyone to comment on one’s post.

3. The already created posts also has a like and dislike feature on it so other users can as well as comment on the post but also like or dislike it, which may spur a discussion, which is another objective of ours however we are just focusing on the ability to like and dislike a post. When another user sees a post and likes it they’re also giving the option to unlike the same post which in turn makes the like for the post go away.

### 2.2 Test Plan

The system is set up to where once a user is logged on and wants to look at any post posted by other users in case they want to view, comment , and/or like or dislike their post. They are able to by clicking on the Blog Posts tab. The starting point of this usability test shall start after clicking onto Blog Posts where the user will be either sent to a login screen or directly to the front page of the Blog Posts section.

The task to be accomplished is after clicking onto the Blog Posts section the user is supposed to be able to view other users posts and once at that point the user is supposed to click on one of the posts and comment on the post and/or like or dislike the post. This will get the user to intergage with this section of the website. The intended user is the person looking to comment or engage with the post made by other users, someone who wants to engage in a conversation with anyone else that favors that topic. This is what gets other users to make their own post to see if there is anyone else that is interested in the same topic as them.

A user knows they have completed this test once they realize that they can view their post on another user's post, when they are able to see that they can like or dislike the post and the ability to change and witness the fact that they are unable to both like and dislike someone's post. Lastly once the user has completed posting their comment they should also notice the fact that they can view more recent user comments on the same post.

URL : https://www.peaceofmind-cen4010.xyz/blog-posts.php

### 2.3 Questionnaire

|  | Strongly Disagree | Disagree | Neutral | Agree | Strongly Agree |
| --- | --- | --- | --- | --- | --- |
| Uploading my post onto PeachOfMind was easy, the ease of access this website gives me compared to its competitors is almost identical. |  |  |  |  |  |
| While on the post page I am able to use the search bar when looking for specific post that I or my follow users may have posted. Which makes it easy if I want to find a specific group that caterers to my views and/or my liking. |  |  |  |  |  |
| Once uploading my post, I was able to view it right away and able to make changes if need be. Other users can view my post and comment their own thoughts onto my post. |  |  |  |  |  |

# 3 QA Test Plan

### 3.1 Test Objectives

In this QA test, our group will be in charge of testing the blog posts section, which is the section where users can see all the blog posts that have been posted on Peace of Mind.

1. Our first objective is to ensure that blog posts are only available to users who have signed in already into Peace of Mind. This ensures that blog post data is only accessible to registered users in our database, and at the same time blog posts must only be created by signed in Administrators which ensures that the information being shared in the blog is secure and appropriate to all the users in the platform.
2. Our second objective is to ensure that when an administrator decides to create a post, the data that they want to share in Peace of Mind is correctly stored into our database. That includes text information such as the title, summary, body, creation date timestamp, and post image.
3. Our third objective is to make sure that all administrators can store multiple blog posts. To do this, each person in the group will share a post.
4. Our fourth objective is to make sure that all the blog posts are displayed correctly and that the regular users can see the posts and access their complete information.
5. Our fifth objective is to make sure that the user has access to a search bar where he/she can narrow down their searches to a specific part of a blog post content, for example the user should be able to look a post by category name.

### 3.2 Hardware and Software Setup

The hardware and software setup of the person in charge of testing must be similar to that of an average Peace of Mind user. They must have access to a desktop or laptop computer or a cellphone where they can see all the blogposts that exist in the database. They must also have access to a modern web browser such as Google Chrome, Mozilla Firefox, or Microsoft’s Edge.

Also, the tester must have access to the hardware and software that an administrator should have, which is a desktop or laptop computer so that he can test the administrator dashboard backend functionality which is where administrators can create and edit posts and categories, as well as to edit the user privileges, such as converting a user into a administrator.

### 3.3 Features to Be Tested

The feature that our team has decided to evaluate during the QA testing is the blog posts section which consists of a screen where all the posts are briefly presented to the user in the form of Bootstrap cards. These cards contain the title, summary, and date of creation of the blog post. Also, the user must be able to access the complete blog post by clicking on the link that will be found in the title of the post. We have also decided to have a search bar which will allow users to narrow down their search. With this context in mind, the features to be tested include making sure all registered users have access to this section of Peace of Mind, then all post summary cards must be correctly fetched from the database and be display in the user’s web browser. These post cards will need to provide access to the correct and complete blog post.

Peace of Mind URL: <https://peaceofmind-cen4010.xyz/>

### 3.4 Actual Test Cases

First Test Case Scenario:

1. The first case scenario will consist of the tester not being logged in into Peace of Mind. Then the tester tries to access the blog posts section. This will result in the user being redirected to the login screen, which also contains a link to the registration page through the navigation bar.

Second Test Scenario

1. In the second test scenario, the tester wil be logged in into their account. The tester will then be able to access the blog posts categories and he will be able to see all the blog posts summaries.
2. The tester shall be able to look at all the existing blog post cards that contain a link to the full post that he selected.
3. The user should be able to view the post full content after clicking the post summary card’s title which contains an anchor to the post.
4. User shall be able to view all comments that exist for that blog post and the number of likes and dislikes.
5. Tester shall be able to see which user made a specific comment for the given post

Third Test Case Scenario:

1. The tester will be signed in into Peace of mind and he accesses the blog posts section through the categories section
2. The user shall be able to see the post summary cards of those post that the tester has selected only.
3. The summary cards shall contain a link that redirects the user to the selected full blog post.
4. Tester shall be able to view the full blog post content after clicking the summary card title
5. Tester shall be able to see all the comments for that specific post as well as the number of likes and dislikes.
6. Tester shall be able to see which user commented the selected post and have a way to access their profile page

Fourth Test Case Scenario

1. The user is signed in into their account and they access the blog posts section of Peace of Mind.
2. The tester inputs in the search bar a query by blog post title, body content, summary, author, category, or date of creation.
3. The search is selected in the blog’s database and the information about the post is displayed into the information cards that are generated and displayed into the user’s screen.
4. Tester shall be able to view the full blog post content after clicking the summary card title
5. Tester shall be able to see all the comments for that specific post as well as the number of likes and dislikes.
6. Tester shall be able to see all comments for the given post and shall have access to the user profiles that have commented that post.

### 3.5 Perform Testing

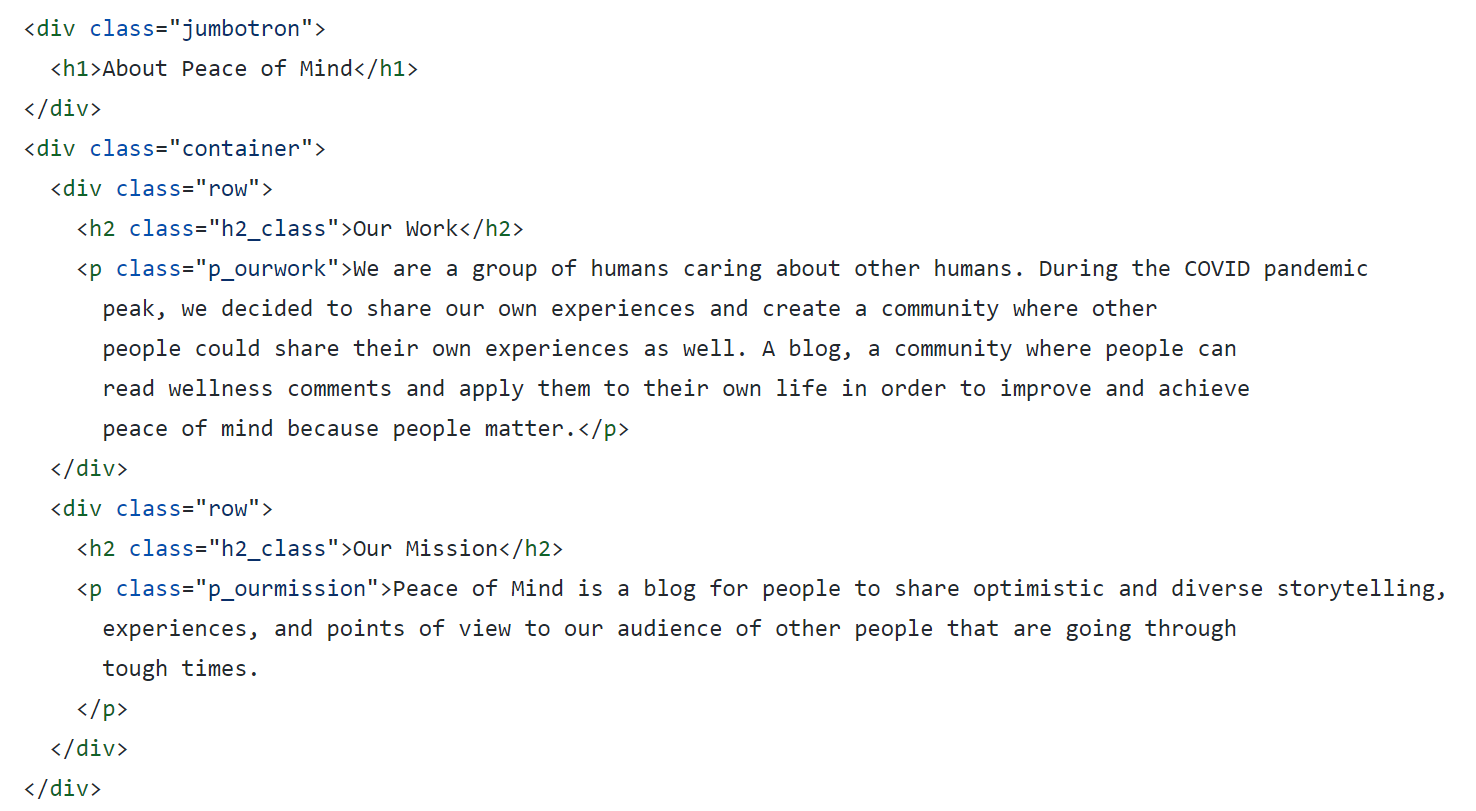
| Test Case | Google Chrome | Mozilla Firefox | Microsoft Edge |
| --- | --- | --- | --- |
| Test Case 1 -User is not signed into their account and are redirected to login. | Pass | Pass | Pass |
| Test Case 2 - User is logged in and access blog post section. | Pass | Pass | Pass |
| Test Case 3 - User is logged in and access blog post section through the blog categories section. | Pass | Pass | Pass |
| Test Case 4 - User access blog post section and uses the search bar functionality. | Pass | Pass | Pass |

# 4 Code Review

The coding style we chose to use for this project was low-level programming. Due to the minimal levels of abstraction required for this assignment, low-level programming was sufficient enough to build a fully-functioning website. Low-level coding languages have a lower level of abstraction, and as you might expect, are the opposite of high-level languages. They are closer to binary and farther from human language. Low-level languages are harder to learn and use, but offer more functionality and direct control over the computer. They allow programmers to create far more detailed and efficient computer software. The code was based in php and used query style language to interact across code segments.



The first code sample above correctly used div containers to separate segments of code. The div container first used at the code top is creating a container for the header segments. The header blog post uses php to query the category of the blog post. One comment for the future correction of the code style would be to apply classes to each code block. For example, if we wanted to style just the blog header separately than the rest of the blog material, we might want to apply a HTML class, ie class=”blogHeader” to separately customize the blog.



As we can see in this second snippet of text, some of the HTML classes are not defined clearly and can create some possible errors in the code at future times of editing. For example, the first class titled “jumbotron” might want a more relatable and identifiable name, such as “mainHeaderText” to quickly identify the placement of that text on the website. The second header we see labeled as “h2\_class” might also want a more recognizable name so that when future developers are developing the site, they can easily distinguish between which h2 is being used on this page versus the headers on other pages.

Overall, the code was well built with minimal errors. The code was concise and did not repeat segments often, a good sign of well built, abstract code. While some of the code was slightly vague, it was still easy enough to parse through and could be understood while reading through the lines of code. Code review is a crucial step of code deployment because it allows the development team to review all functioning parts of a product or software to ensure that the product that reaches consumers hands are clean to operate and smooth for users to interact with.

# 5 Self-Check: Security

### 5.1 Assets to Protect

The user’s password shall be protected by encrypting the password key. Assets such like this and others like username, images, are stored in the database. To prevent this, it is important to sanitize the data inputs and to ensure that proper security is placed to prevent brute-force login attempts.

### 5.2 Confirmation of password encryption

It is possible for the cyber criminal to brute force the login system and gain access to usernames and passwords that are stored in the database (via SQL injection). Passwords are encrypted to ensure that if the data is leaked, it will take some time to decrypt the string that hides the password.

### 5.3 Confirm input data validation

Any data that is needed to be input in the search bar, comments section, login system shall sanitize the user input to prevent unwanted access to the website.

# 6 Self-Check: Adherence to Original Non-Functional Specs

*Performance* - Web pages must be optimized to be loaded in less than 300ms but no more than 1 second. Of course, internet speed may vary. A slower connection will take a long time to load the page. This product should be a light application that is able to produce a light load to the system. The minimum requirements are Windows 8 Operating System with 4GB RAM and the Recommended requirements is Windows 10 with 8GB RAM. A video card is recommended with at least 2GB VRAM to make use of all the applications of the system. The response time between users for the live chat should take less than 200ms but no more than 500ms. - DONE

*Portability* - Webpage shall be designed to be accessed from any web browser supported on any operating system. For example, all users should be able to access the webpage from Windows 10, Mac OS, Linux, etc. and is accessed from any web browser such as Internet Explorer, Google, Bing, Firefox, etc. Any plugins used for the website should be compatible with every operating system and with every web browser available to the user. - DONE

*Usability* - Webpage shall be designed to have easy functionality of the website. The interface of the website shall be easy to use and easy to navigate through. High contrast colors should be use to allow the user to see properly. - DONE

*Accessibility* - Webpage shall be created to accommodate those with disability. Webpage shall have options for those who are hearing impaired, sight impaired or for any other disability. High color contrast shall be used for those that are sight impaired. Text-to-speech technology should also be implemented for those who are sight impaired. - DONE

*System load* - Webpage shall be optimized to induce minimal load to a system. Webpage functionality shall support up to 1000s of people using the service. Of course this depends on the location. A smaller population will see less load on the webpage while a larger population will see more load to the webpage. Webpage shall be optimized to ensure that services are not significantly slowed down or completely stopped. - DONE

*Security Requirements* -

* Regular users shall not have access to other user’s passwords. Users shall not have access to administrative rights and they shall not have access to client information except for themselves.
* Administrators shall have access to the client and server side of the product. Administrators shall monitor web pages and make sure that there are no vulnerabilities in the website and monitor that the content posted by users is appropriate for the blog’s theme.
* Passwords shall be encrypted to make it more difficult for the attacker to access it. The blog shall protect itself from SQL injection by using prepared statements for all the text input by the user.
* Webpage shall include a password limit where if the user tries to enter their password more than 3 times, the webpage will lock them out and they will have to wait 24 hours to try again. This is to prevent brute-force attacks on the system.
* Website shall not save anything on a text file.
* Website shall save the data onto a database.
* Users shall have to verify their email address before creating an account.

- DONE

*Storage* - All data must be stored in a database and vital information must be encrypted.

- DONE

*Availability* - Webpage must be available 24/7 for 365 days (or all year long). - DONE

*Maintainability* - A webpage that is running until the end of time shall be subjected to maintenance, therefore it is imperative that downtime shall be enforced to update or to fix issues regarding the webpage. If it is for maintenance, downtime shall be less than 30 minutes but no more than 1 hour. If there are any other issues, downtime shall be less than 6 hours but no more than a day, depending on how serious the issue is. - DONE

*Fault tolerance* - In the event of webpage failure (such data breach, etc.) webpage shall be temporarily taken down to resolve issues and to further prevent attacks to the system. Administrators shall implement policies and procedures in time of a cyber attack. Administrators/Owners of the website shall notify all users if data has been breached to be in compliance with GDPR laws and regulations of other countries or states. - DONE

*Privacy* -

* User information shall be hidden and protected from outside sources and their information shall not be sold to third party sources.
* Each user is responsible for the protection of their personal information.
* Users shall have the right to request and download data gathered from the webpage. Users shall also have the right to request termination of their account.
* Users shall have the right to make their information either private or available to the public.

*Quality* - Webpage shall be designed to be up to standards in terms of web application development. Webpage shall provide a user interface that is easy to use and easy to learn. Webpage shall not be bloated with unnecessary features. Webpage shall be scaled to fit devices. For example, a web page should be scaled to fit 1920 x 1080 resolution, 2k resolution, 4k resolution, etc. Ideally, any resolution shall not detract the aesthetic quality of the website, no matter what device it is being accessed from. - DONE

*Support* - Webpage shall have support running 24/7 for 365 days (or all year long). It is the administrator/Owners responsibility to acknowledge and try to alleviate the issue to the best of their ability. Users shall have the ability to submit any tickets. - DONE

# 7 Submission: Youtube Video



Video URL: <https://youtu.be/JwIZyYuxxgA>

# 8 GitHub Link:

Github Repository Link: <https://github.com/CEN4010-F2021-g3/cen4010-group3>